

Compressive Single-Pixel Imaging

Andrew Thompson

School of Mathematics and the Maxwell Institute
University of Edinburgh, Edinburgh, EH9 3JZ
a.thompson-8@sms.ed.ac.uk

I. INTRODUCTION

For imaging in the visible spectrum, CCD and CMOS technology has allowed effective and compact airborne sensing devices to be developed for the defence industry. These devices typically acquire pixel samples by means of a full array of photon detectors. However, imaging in other wavelengths can require the need for more exotic detectors, and a vast array of these expensive and bulky detectors may no longer be feasible. This motivates the search for alternative imaging techniques which are capable of delivering good image quality while reducing the cost and bulkiness of the sensing device. Furthermore, there are computation and storage issues: the amount of data generated from a full set of pixel samples may be so large that, for storage or transmission purposes, the data must be compressed. The traditional approach, in which a full set of measurements is acquired, only for much of the information to be subsequently discarded, appears intuitively wasteful.

The emerging theory of Compressed Sensing (CS) offers a potential solution to all these issues [3]. It asserts that compressed images can be recovered from a significantly undersampled set of measurements, provided two conditions are met. Firstly, the image must be suitably compressible in some transform domain. Secondly, an appropriate randomized sampling scheme must be used. Central to the theory is the design of suitable CS algorithms to recover the image from the undersampled measurements. This paper concerns a new approach to single-pixel imaging which exploits CS theory. The camera design in question was first proposed by a team at Rice University [4], and they built a proof-of-concept model for visible light.

The aim of this paper, as well as providing some background on the theory of Compressed Sensing and the CS single-pixel camera, is to report the results of some numerical experiments conducted on a computer model of the camera design, and to highlight the scope and potential of this new technology. The structure of the paper is as follows: In Section II we give a description of the camera design, before describing in Section III how

a simple mathematical model of the camera fits naturally into the CS framework. Section IV describes some CS recovery algorithms for this problem, and we report the results of some numerical tests in Section V.

II. THE CS SINGLE-PIXEL CAMERA

In contrast to a conventional camera which would have a vast array of photon detectors – one for each pixel – a single-pixel camera is so-called because it has only a single photon detector or ‘pixel’. The incident light field is directed onto a specialized type of spatial light modulator known as a Digital Micromirror Device (DMD) which consists of an array of tiny mirrors, one corresponding to each pixel. Each mirror can be oriented in one of two directions: the ‘on’ position directs the light for that pixel towards the single detector, while the ‘off’ position directs the light away from the detector. The light from all the pixels set to the ‘on’ position is then summed and a measurement is recorded as an output voltage on the photon detector. The measurement is then subsequently digitized by an A/D convertor. A series of measurements can be obtained by flipping the mirrors and repeating the process a number of times. See Figure 1 for a diagram of the camera design. Clearly, the result of such a procedure is encoded measurements, and the other issue to address is how to decode these measurements and recover the incident image.

The concept of a single-pixel camera is not a new one, and it fits into the broad category of multiplexing imaging methods in which a series of consecutive measurements are made by a single detector. What sets the CS single-pixel camera model apart is a novel sampling approach which means that it is possible to take fewer measurements. It therefore offers an alternative approach to obtaining compressed images: rather than taking a full set of samples and subsequently compressing, compressed samples are acquired in the first place. To achieve this, CS theory motivates the use of a random sampling procedure in which each mirror is set randomly to either the ‘on’ or ‘off’ position with equal probability.

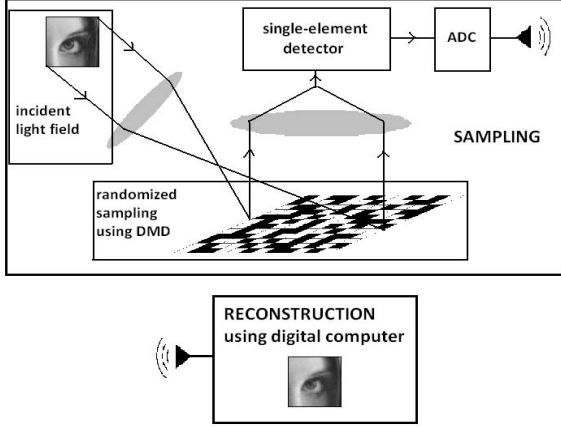


Fig. 1. A schematic diagram of the Rice CS single-pixel camera.

III. A MATHEMATICAL MODEL

Modelling the incident light field as a discrete pixelated array consisting of N pixels, we may also represent this ‘original image’ as a vector $x \in \mathbb{R}^N$. By means of the DMD, the light from all the pixels set to the ‘on’ position is summed and recorded as a voltage on the photon detector. We may model this summation as an inner product of the original image x with a test function $\phi_i \in \mathbb{R}^N$ consisting of random ones and zeros, giving a single measurement $y_i \in \mathbb{R}$. Sampling error is likely to occur, particularly as a result of photon counting noise, and subsequently due to quantization error in the digitization. We choose to adopt a simplistic model for sampling noise in the form of additive Gaussian white noise. We may therefore write

$$y_i = \langle x_i, \phi_i \rangle + e_i,$$

where $e_i \sim N(0, \sigma^2)$ for some noise parameter σ .

In keeping with the CS framework introduced in Section I, we choose to undersample the image and take n such measurements, where $n < N$. Writing Φ for the $n \times N$ matrix whose rows are the test functions $\{\phi_i\}$, we may represent the entire sampling process by the matrix equation

$$y = \Phi x + e,$$

where $y \in \mathbb{R}^n$ is the vector of samples. The sampling matrix Φ , as introduced here, is therefore a random Bernoulli matrix consisting of equiprobable ones and zeros. The DMD would also permit the use of other sampling matrices such as random ± 1 entries, which is what we used in our numerical tests since it leads to better performance of the recovery algorithms.

The samples are then sent to a digital computer for decoding. As was explained in Section I, recovery is

based upon the assumption that the original image is compressible in some transform domain, so that it can be well-approximated by a sparse vector in that domain. In this regard, let Ψ be an $N \times N$ transform matrix – in our numerical tests in Section V we will use wavelet transforms – and suppose that $z = \Psi x$ has exactly k non-zero coefficients, where we assume $k < n < N$. We will refer to such a vector as being k -sparse. We can now frame the reconstruction task as an optimization problem. To do this, observe that we have two goals to achieve: we wish to obtain a close approximation to the linear system $y = \Phi x$, while we also wish to obtain a k -sparse solution. This can be achieved by solving the optimization problem

$$\min_{x \in \mathbb{R}^N} \frac{1}{2} \|y - \Phi x\|_2^2 \quad \text{subject to} \quad \|\Psi x\|_0 \leq k, \quad (1)$$

where we minimize an l_2 -norm objective function to achieve a good fit to the linear system of measurements, and where we use an l_0 -norm constraint to control the sparsity (the l_0 -norm simply counts the number of non-zero coefficients of a vector). We turn next to the remaining challenge, which is to identify recovery algorithms for solving this problem which, on the face of it, appears to be combinatorial in nature and therefore far from straightforward.

IV. RECOVERY ALGORITHMS

The CS single-pixel camera may be modelled as a two-stage process of sampling followed by recovery. We built a MATLAB implementation of both stages of this process, which enabled us to carry out a range of numerical tests, the results of which are the subject of Section V.

The model was implemented with a variety of choices of sparsifying transform. We made use of various industry-standard transforms, including the Discrete Cosine Transform (DCT), Haar wavelets and Daubechies 9-7 biorthogonal wavelets. The DCT is the basis of the JPEG compression standard, and the Daubechies wavelet is one of the transforms used by JPEG-2000. Daubechies wavelets are generally a good option for piecewise smooth images, while Haar wavelets represent an alternative for naturally angular images. The various options allow an appropriate choice of sparsifying transform to be made for the image in question. For further details on sparsifying transforms and wavelets, we refer the reader to [6].

The model also included a choice of three CS recovery algorithms. All the algorithms implemented in the model are gradient projection algorithms either for (1), or for a variant of (1).

- **Normalized Iterative Hard Thresholding (NIHT).** This algorithm [2] proceeds by means of the iteration

$$x^{m+1} = H_k\{x^m + \alpha^m \Phi^T(y - \Phi x)\},$$

where α^m is a step-size, and where H_k is the Hard Threshold operator which projects onto the set $\{\|\Psi x\|_0 \leq k\}$. In fact, H_k is equivalent to simply keeping the k largest coefficients in the Ψ domain, and setting the rest to zero. Note that $-\Phi^T(y - \Phi x)$ is the gradient of the objective in (1). We employed the normalized step-size scheme proposed in [2], which guarantees stability of the algorithm.

- **l_1 -Projection.** A common strategy in CS is to replace the l_0 -norm with the l_1 -norm, motivated by the fact that minimizing the l_1 -norm also promotes sparsity, with the additional benefit that the problem becomes convex. This particular l_1 -based approach is very similar to one proposed as part of the *SPGL1* code [8]. More precisely, we replace the Hard Threshold H_k with P_k , the projection onto $\{\|\Psi x\|_1 \leq \tau\}$, for some parameter τ .
- **Iterative Tree Projection.** It is possible to consider more refined models of sparsity. In particular, large wavelet coefficients tend to have a connected tree structure which can be imposed upon the solution. It is in fact possible to project onto the set of vectors whose wavelet coefficients are supported on connected trees of size k , which gives a third alternative [1].

V. NUMERICAL RESULTS

As well as the options described in Section IV, there are also three parameters that the model user is able to control.

- **Undersampling ratio.** Given an input image of dimension N , we take n samples, and so the ratio $\delta = n/N$ gives the level of undersampling.
- **Tuning parameter.** Each algorithm requires the input of a tuning parameter which either directly or indirectly controls the sparsity of the recovered solution in the transform domain.
- **Noise level.** Noise in the sampling process is modelled as additive Gaussian white noise, the volume of which may be controlled by the parameter σ , so that noise drawn i.i.d. from a $N(0, \sigma^2)$ distribution is added to each measurement.

A key aim of the experimentation carried out was to understand the interplay between these parameters, and how they affect performance metrics such as recovery accuracy. In particular, the crucial practical question is

how the recovery accuracy is impacted by reducing the number of measurements, i.e. decreasing the undersampling ratio δ .

A. Systematic testing of the parameter space

If we make the simplifying assumption that there is zero sampling noise, there are two parameters that we are free to vary: the undersampling ratio δ and the tuning parameter (k or τ). We may therefore test the model on a mesh of equally-spaced grid-points throughout the parameter space, and use this to identify the optimal parametric configuration for the test data. For each point on the mesh, 100 trials were performed and the results averaged. In each trial, the model is applied to the same 64x64 cut from lena (showing the recognizable feature of an eye).

Figure 2 shows a plot of average RMSE throughout the parameter space for ± 1 sampling, the l_1 -projection algorithm and 9-7 wavelets. The tuning parameter τ is here normalized as a factor of the l_1 -norm of the original image in the transform domain, which we refer to as the τ -factor. A key feature, which was in fact observed across the whole range of model options, is the existence of an ‘optimal’ tuning parameter for a given level of undersampling, i.e. the value of the τ -factor which minimizes the RMSE.

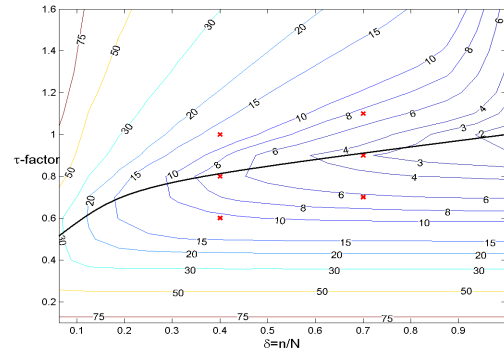


Fig. 2. Average RMSE for l_1 -projection.

The choice of tuning parameter represents one of the challenges for implementing CS algorithms, since the sparsity or compressibility of the scene in question may not be known *a priori*. In practice, known results for test images, together with knowledge about the nature of the scene in question, may help to generate ‘rules of thumb’ for choosing the tuning parameter. On the positive side, the tuning parameter may be viewed as giving the user the flexibility to choose the desired compression ratio. It is also worth pointing out that, due to the nature of

the recovery algorithms, it would not be too costly to generate solutions for a sequence of values of the tuning parameter.

The superimposed black curve traces out the optimal tuning curve. Following the optimal tuning curve through the parameter space from right to left, we see that the solution RMSE gradually degrades as the undersampling ratio is decreased. In other words, we would expect a controlled degradation in recovery accuracy as the number of samples is reduced.

To illustrate these observations, Figure 3 gives six recovered images corresponding to the points marked with red crosses on the RMSE plot. Two different undersampling ratios ($\delta = 0.4$ and $\delta = 0.7$) have been selected, and for each we provide an example at optimal tuning, below-optimal tuning and above-optimal tuning.







	$n/N = 0.4$ ($n = 1638$)	$n/N = 0.7$ ($n = 2867$)
above-optimal tuning	 RMSE : 13.46 time : 3.58s tuning : 1.0	 RMSE : 7.53 time : 6.17s tuning : 1.1
optimal tuning	 RMSE : 7.10 time : 5.01s tuning : 0.8	 RMSE : 3.00 time : 9.20s tuning : 0.9
below-optimal tuning	 RMSE : 10.53 time : 1.16s tuning : 0.6	 RMSE : 6.21 time : 1.86s tuning : 0.7

Fig. 3. Examples of images recovered by l_1 -projection.

Data obtained on algorithm running time was also promising: we found that, for these problems, and on a workstation consisting of two 6-core processors, running times were everywhere under 5 seconds.

B. Robustness to noise

We also investigated the impact of the injection of additive Gaussian white noise e such that $y = \Phi x + e$ and each entry of e is distributed as $N(0, \sigma^2)$. We did this by determining by what factor the RMSE of the noise is amplified in the recovered solution. Identifying this factor, however, is complicated by the fact that we have recovery error due not only to the noise, but also due to natural compression loss. In order to isolate the contribution from the noise alone, we first compressed the 64x64 lena ‘eye’ test image so as to be k -sparse in the 9-7 wavelet domain, where k is the sparsity tuning parameter for the NIHT algorithm. CS theory tells us that, in the absence of noise, such an image can be exactly recovered with high probability within a certain region of the phase space. Any recovery inaccuracy can

therefore be fairly attributed to the presence of noise in the sampling process. Given a recovered solution \hat{x} , we therefore define the noise amplification factor to be the solution RMSE divided by the RMSE of the noise vector e , i.e.

$$AMP = \frac{\sqrt{\frac{1}{N} \|\hat{x} - x\|_2^2}}{\sqrt{\frac{1}{n} \|e\|_2^2}}.$$

Figure 4 gives the noise amplification factor plot for $\sigma = 2.5$ (approximately 1% of the dynamic range of the image). Superimposed over the plot is the optimal tuning curve. We see that in the region of the optimal tuning curve, the noise amplification factor is well-behaved and in fact never exceeds 2. In fact, the amplification factor can be significantly lower even than this, especially for small δ . In other words, the CS recovery can even have a denoising effect, actually reducing the RMSE. An explanation for this behaviour is that, since the CS recovery obtains a good approximation to the best classical compression in which small coefficients are thresholded out, the result is that small noise contributions can be thresholded out as well.

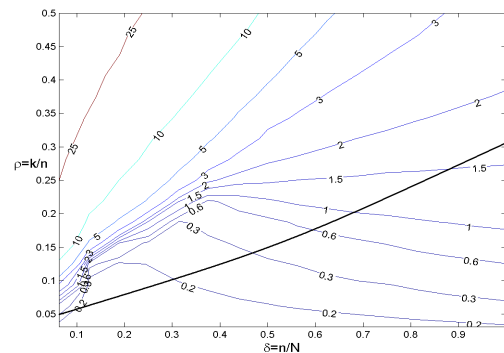


Fig. 4.

Noise amplification factor for NIHT.

C. Multi-spectral imaging with clutter

While the single-pixel camera was first studied in the context of visible light, it is for other wavebands, such as infra-red, that it is more likely to find an application. Sampling light outside the visible spectrum often requires the use of more exotic detectors which may be either expensive or bulky. In such cases, there may be considerable incentives to explore the option of moving from many photon detectors to a single detector or ‘pixel’. One possible application of the CS single-pixel approach is in multi-spectral imaging in which the light field would be simultaneously directed (by means

of an appropriate DMD configuration) onto a number of detectors corresponding to different wavebands. It is interesting therefore to address the issue of how recovery of the same scene may vary across different bands.

Furthermore, objects of interest are in reality likely to be surrounded by clutter, either in the foreground or background, and a crucial issue for any imaging approach is the dependency of recovery accuracy upon the level of such clutter. To investigate this, we used the *CAMEO-SIM* package [5] to simulate images of the same scene, in three different wavebands, with and without background clutter in the form of trees. Figure 5 shows the recovered images, with and without trees, for three infra-red wavebands. The model options were ± 1 sampling, the l_1 -projection algorithm, Haar wavelets, and undersampling ratio 0.15.













		without trees		with trees		
	data	original	recovered	recovered	original	data
SW	RMSE : 9.13 t : 72.57s tuning : 0.8					RMSE : 26.87 t : 26.86s tuning : 0.4
MW	RMSE : 11.37 t : 54.24s tuning : 0.6					RMSE : 18.78 t : 55.14s tuning : 0.5
LW	RMSE : 8.60 t : 52.60s tuning : 0.7					RMSE : 15.97 t : 34.33s tuning : 0.45

Fig. 5.

Example of recovery in different wavebands, with and without background clutter.

D. Dynamic imaging

So far we have assumed that the scene of interest is static over the period of the image acquisition. A natural next question to ask is whether these CS techniques can be extended to moving images. This question has already been addressed in the CS literature, see for example [9].

Traditional CS imaging theory considers a series of measurements taken from a single image. However, for a moving scene, each measurement will act on essentially a different image. To be able to leverage traditional CS imaging theory, therefore, we must follow [9] in making the modelling assumption that the image changes slowly over a group of snapshots, which we can then equate to a single video frame. Under this assumption, we may represent an acquired video as a sequence of F frames, each consisting of N pixels, where we take n measurements per frame. Omitting sampling noise from our consideration for simplicity, we may therefore

model the acquired measurements as $y_i = \Phi_i x_i$ for $i = 1, \dots, F$, where each Φ_i is an $n \times N$ sampling matrix. A basic approach is now to view the problem as F separate CS problems. Selecting Ψ to be some appropriate 2D sparsifying transform, we may then apply any of the recovery algorithms from Section IV to each frame.

However, an alternative approach is possible in which we attempt to exploit the temporal dependence between frames. To do this, we consider the entire video sequence, and represent it as a single vector

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_F \end{bmatrix}$$

of length NF . The sampling process may now be written in the form of a single acquisition as

$$y = \Phi x = \begin{bmatrix} \Phi_1 & 0 & \dots & 0 \\ 0 & \Phi_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Psi_F \end{bmatrix}.$$

We may now choose Ψ to be a 3D sparsifying transform and apply any of the recovery algorithms from Section IV to recover the entire video sequence.

CAMEO-SIM was used to simulate a moving image, based upon the image of the vehicle against a clutter-free background introduced in Section V-C. The temporal dependence between the slides is well-structured: the vehicle moves from the left to right of the shot at constant speed. The video sequence consists of 64 frames where each frame is a 64×64 pixel image, giving a $64 \times 64 \times 64$ datacube. Both frame-by-frame and joint recovery were performed, using the ± 1 sampling scheme, the l_1 -projection algorithm and Haar wavelets (either 2D or 3D). The undersampling ratio was set to $\delta = 0.4$, and the tuning parameter was optimized in terms of the solution RMSE of the entire datacube.

Results for three selected frames are shown in Figure 6. There is clear evidence from the RMSEs that the joint approach gives a more accurate recovery than the frame-by-frame approach. Indeed, the difference may be observed visually: more detail, both of the vehicle and the background, is recovered using the joint approach.

A clear message emerges from this experiment: it is possible to extend the CS approach to moving scenes, provided we can model the scene as a series of static frames. More than this, we can actually do even better by exploiting temporal dependency by means of 3D transforms.










frame	original	frame-by-frame reconstruction	joint reconstruction
24		 RMSE: 7.92	 RMSE: 4.99
32		 RMSE: 9.06	 RMSE: 6.27
40		 RMSE: 8.33	 RMSE: 5.92

Fig. 6.
Comparison of selected frames for frame-by-frame and joint
3D recovery.

VI. CONCLUSION

The results of testing on the proposed camera model are promising in a variety of areas, including single-band and multi-spectral imaging, and 3D dynamic imaging. Furthermore, we observe robustness in the sense of a controlled degradation in image quality in the presence of sampling noise and foreground/background clutter. Further details can be found in [7], which fully documents this work.

VII. ACKNOWLEDGMENT

This work was the outcome of an internship project carried out at *SELEX Galileo, Edinburgh* under the supervision of David Humphreys and Robert Lamb. The project, which was partly funded by the *Underpinning Defence Mathematics (UDM)* scheme, was facilitated by the *Knowledge Transfer Network (KTN)*, and was a collaboration between *SELEX Galileo* and the University of Edinburgh. I benefited from the guidance of my academic supervisors, Coralia Cartis and Jared Tanner, and also David Allwright in his role as *Technology Translator* for the *KTN*.

REFERENCES

- [1] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Trans. in Inform. Theory*, 56:1982–2001, 2010.
- [2] T. Blumensath and M.E. Davies. Normalized iterative hard thresholding: guaranteed stability and performance. *IEEE J. of Selected Topics in Sig. Proc.*, 4(2):298–309, 2010.
- [3] E. Candès and M. Wakin. An introduction to compressive sampling. *IEEE Sig. Proc. Magazine*, 25(2):21–30, 2008.
- [4] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Sig. Proc. Magazine*, 25(2):83–91, 2008.
- [5] M. Gilmore, I. Moorhead, D. Oxford, T. Liddicoat, D. Filbee, C. Stroud, G. Hutchings, and A. Kirk. Cameo-sim: a broad-band scene generation system that is ‘fit for purpose’. *Proc. SPIE*, 3699:217–228, 1999.
- [6] S. Mallat. *A wavelet tour of signal processing, 3rd edition, The Sparse Way*. Academic Press, 2008.
- [7] A. Thompson. Compressive single-pixel imaging. Technical Report ERGO 11-006, School of Mathematics, University of Edinburgh, 2011. <http://www.maths.ed.ac.uk/ERGO/>.
- [8] E. van den Berg and M. Friedlander. Spg11: A solver for large-scale sparse reconstruction. <http://www.cs.ubc.ca/labs/scl/spg11/>, 2009.
- [9] M. Wakin, J. Laska, M. Duarte, D. Baron, S Sarvotham, K. Kelly, and R. Baraniuk. Compressive imaging for video representation and coding. In *Proc. Picture Coding Symposium (PCS)*, Beijing, China, April 2006.